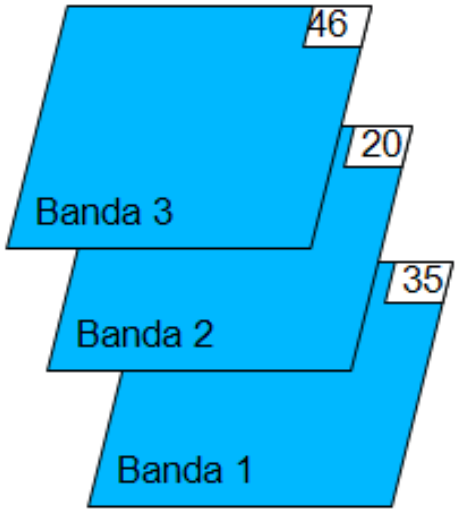
Reconhecimento de padrões

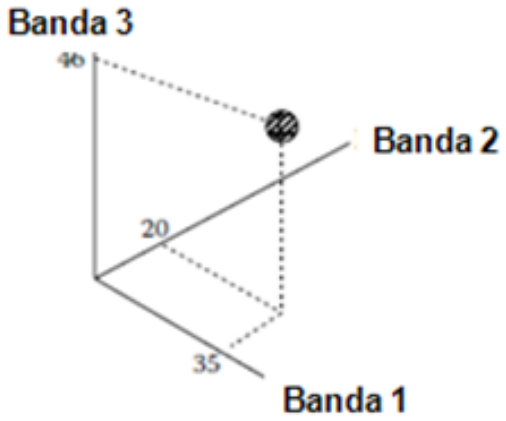
- Classificação não paramétrica:

  - Árvores de decisão
  - Regras de decisão fuzzy

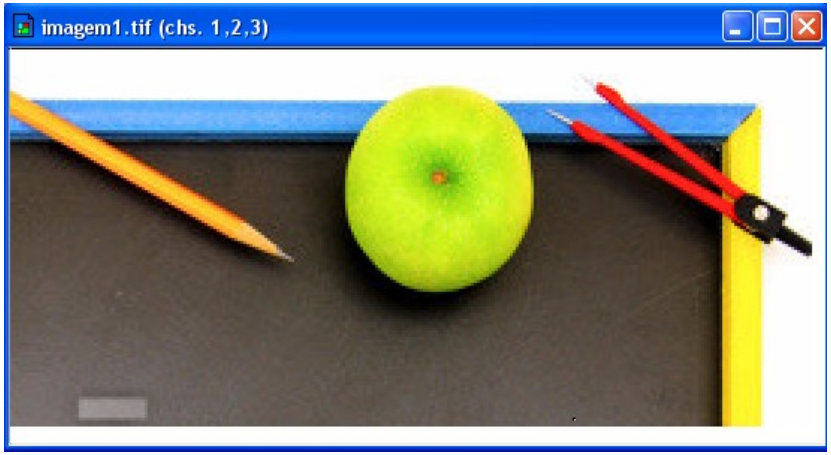


Representação do pixel (i,j) no espaço de características:

$$X_{ij} = \begin{pmatrix} 35 \\ 20 \\ 46 \end{pmatrix}$$

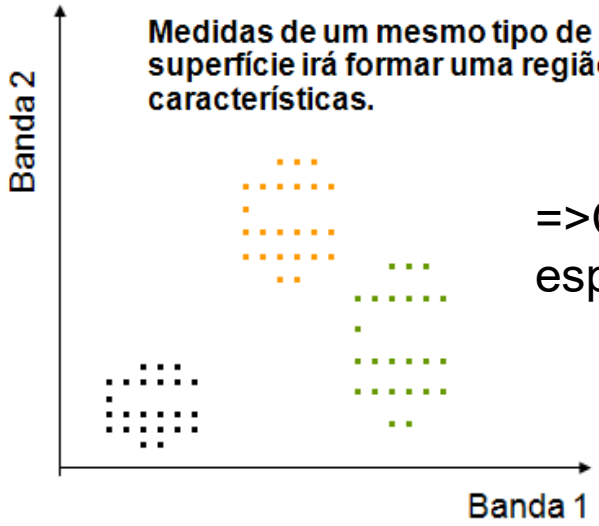


Espaço de características tridimensional



Espaço de características bidimensional:

Medidas de um mesmo tipo de cobertura da superfície irá formar uma região no espaço de características.

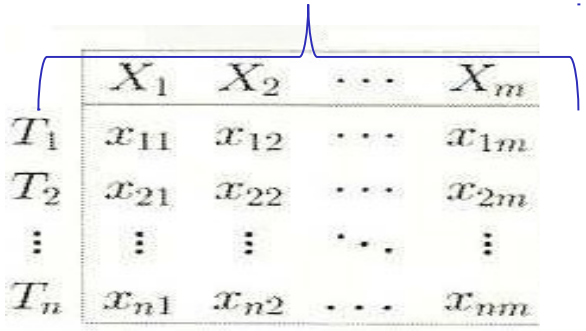


=>Classes espectrais

Aprender como?

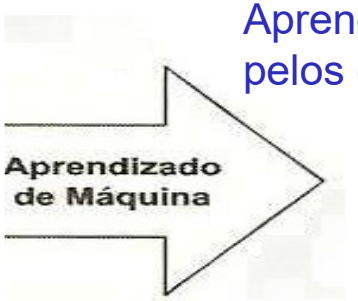
- Dados alguns **exemplos** de um determinado conceito, tentar **inferir** uma definição que permita ao aprendiz reconhecer futuras instâncias daquele conceito.

Atributos ou características descritoras do conceito



$$\begin{array}{c|cccc} & X_1 & X_2 & \cdots & X_m \\ \hline T_1 & x_{11} & x_{12} & \cdots & x_{1m} \\ T_2 & x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_n & x_{n1} & x_{n2} & \cdots & x_{nm} \end{array}$$

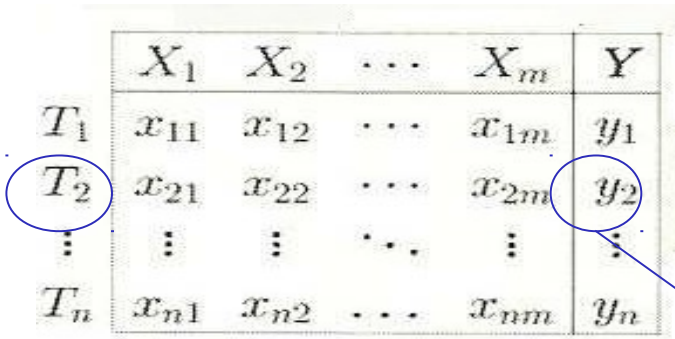
Aprendizado não supervisionado:
Aprendizado dirigido pelos dados



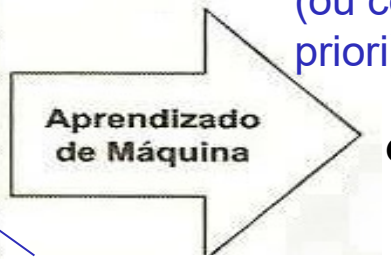
conceito

Aprendizado supervisionado
(ou com uso de conhecimento a priori)

instância



$$\begin{array}{c|cccc|c} & X_1 & X_2 & \cdots & X_m & Y \\ \hline T_1 & x_{11} & x_{12} & \cdots & x_{1m} & y_1 \\ T_2 & x_{21} & x_{22} & \cdots & x_{2m} & y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ T_n & x_{n1} & x_{n2} & \cdots & x_{nm} & y_n \end{array}$$



conceito

Conceito a ser aprendido

- Uma vez aprendido o conceito, o sistema deve reconhecer/classificar instâncias ainda não observadas.

Atributos

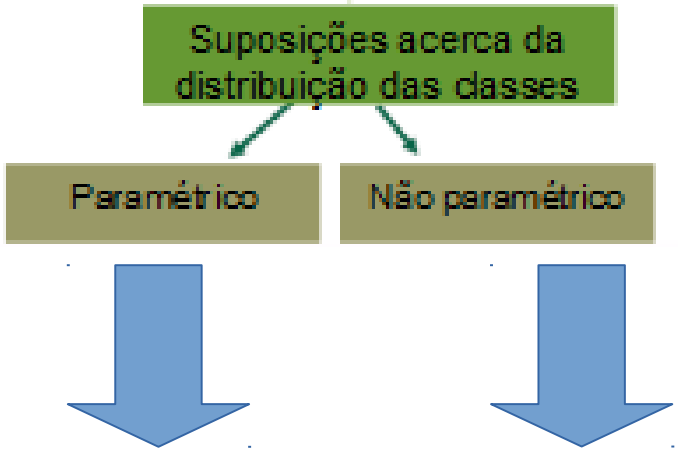
Tamanho	Forma	Furos
grande	longa	0
grande	longa	0
pequeno	longa	0
pequeno	compacta	0
grande	longa	1
pequeno	compacta	1
pequeno	longa	1
pequeno	compacta	1
pequeno	compacta	1
grande	longa	2
grande	outra	2
pequeno	outra	2

Instância →

Atributos

Tamanho	Forma	Furos	Objeto
grande	longa	0	lápis
grande	longa	0	lápis
pequeno	longa	0	parafuso
pequeno	compacta	0	parafuso
grande	longa	1	chave
pequeno	compacta	1	porca
pequeno	longa	1	chave
pequeno	compacta	1	porca
pequeno	compacta	1	porca
grande	longa	2	tesoura
grande	outra	2	tesoura
pequeno	outra	2	chave

Conceito



- Suposição acerca da distribuição estatística das classes
- Restrição na classificação de dados com diferentes escalas de medida e diferentes unidades
  → não permitem integração de dados tais como layers de mapa ou modelos digitais de elevação na classificação

Ex. Classificador da máxima verossimilhança

- Não necessitam de suposições acerca da distribuição estatistica das classes

-Permitem integrar dados com diferentes escalas de medida e diferentes unidades

- as árvores de decisão particionam o conjunto de dados em subconjuntos homogêneos

Ex. Árvores de decisão, redes neurais, etc.

Árvore de decisão

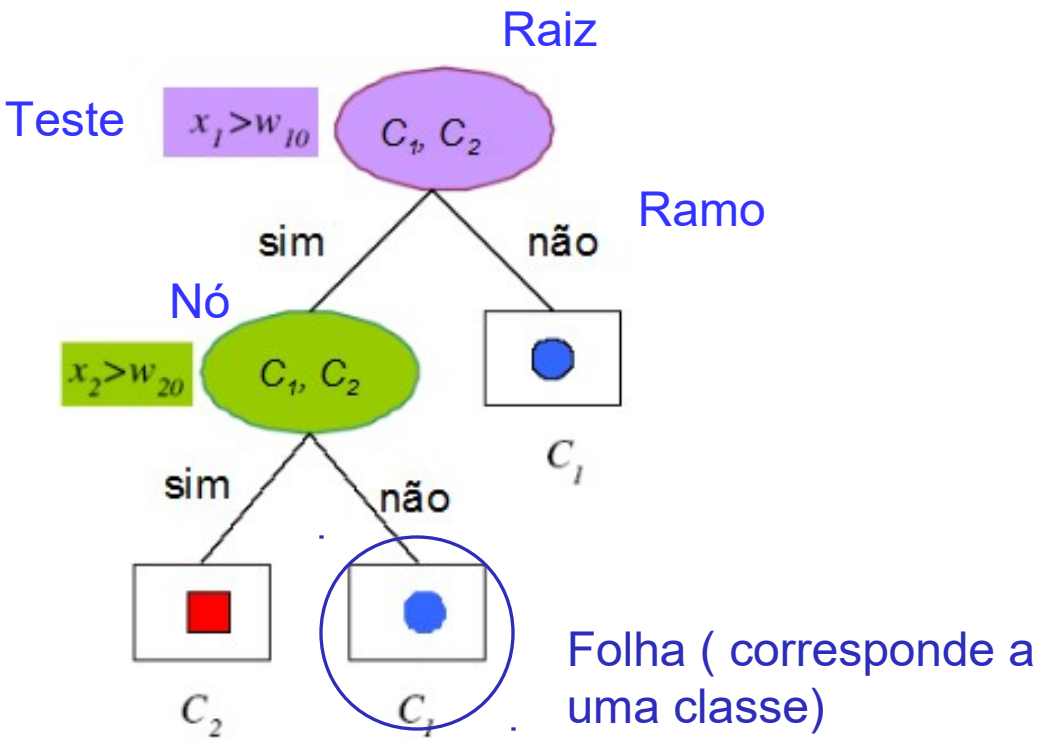
A árvore consiste de:

- Raiz

- Ramos

- Nós

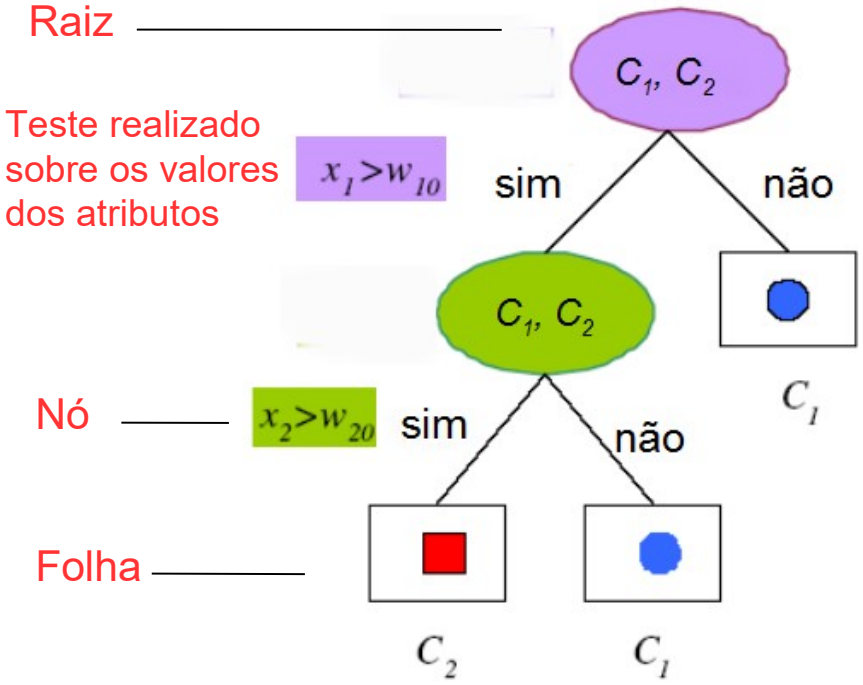
- Folhas



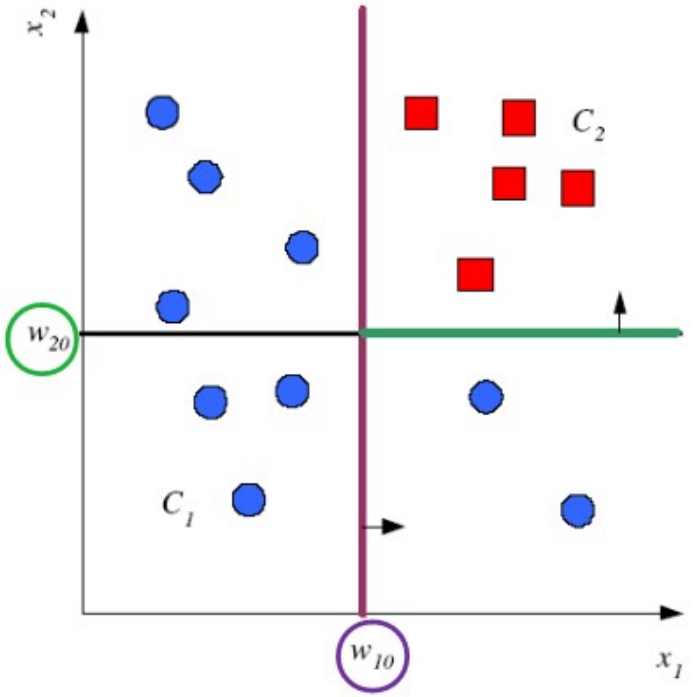
Na *raiz* e em cada *nó* é realizado o teste de algum *atributo,* e cada ramo descendente daquele nó corresponde a um dos possíveis valores para este atributo.

As árvores de decisão particionam o conjunto de dados em subconjuntos homogêneos.

-Interpretação geométrica das árvores de decisão:



Árvore de decisão



Espaço de atributos bidimensional com duas classes

Ex. Parte de uma árvore de decisão construída por meio de um algoritmo de indução de árvores de decisão.

-aprendizado supervisionado

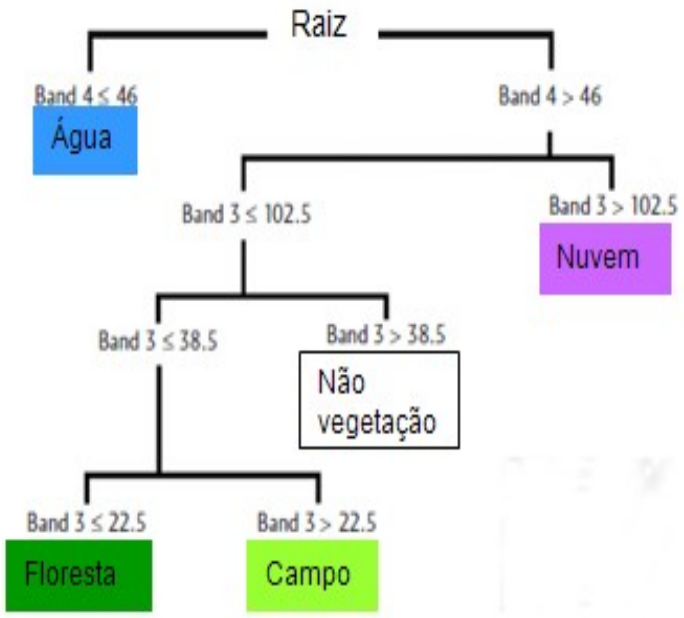
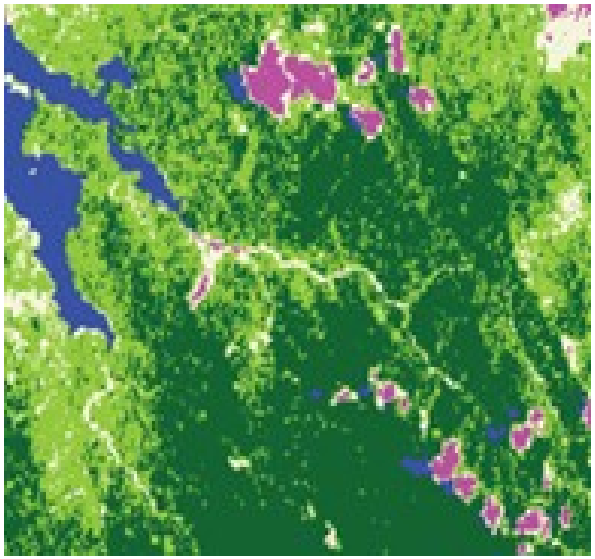


Imagem classificada

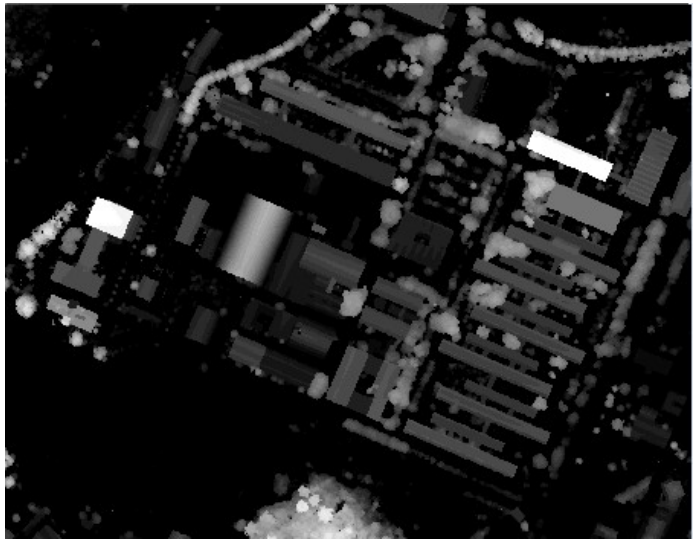


Exercício da aula anterior:

Considere os seguintes dados:

-NDVI, índice normalizado de diferença de vegetação, gerado a partir de uma imagem Quickbird ortorretificada,

-MDSn, modelo digital de superfície normalizado.

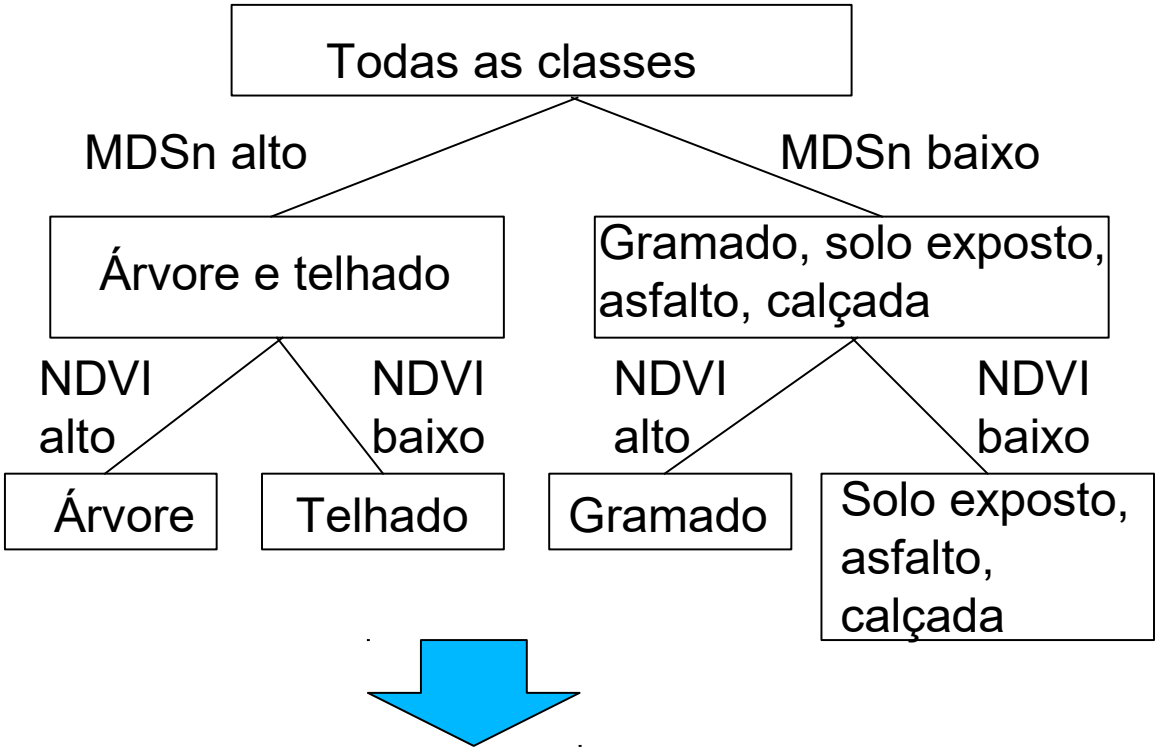


oclusão

MDSn : resolução espacial de 0.7m ; 8 bits.
NDVI : resolução espacial de 0.7m ; 8 bits, com valor 0 para área de oclusão.

→ Testes indicaram que valores de NDVI correspondem à vegetação.

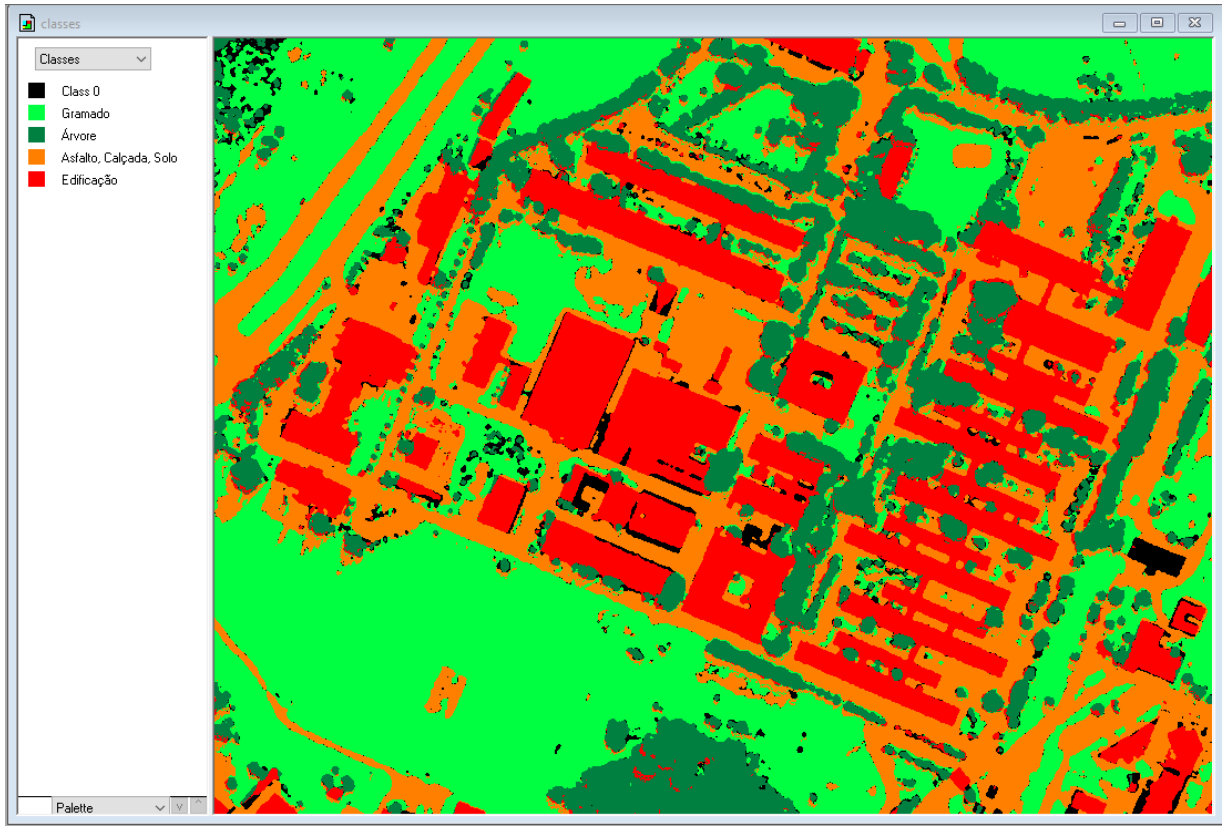
Árvore de decisão:



Regras de decisão para efetuar a classificação

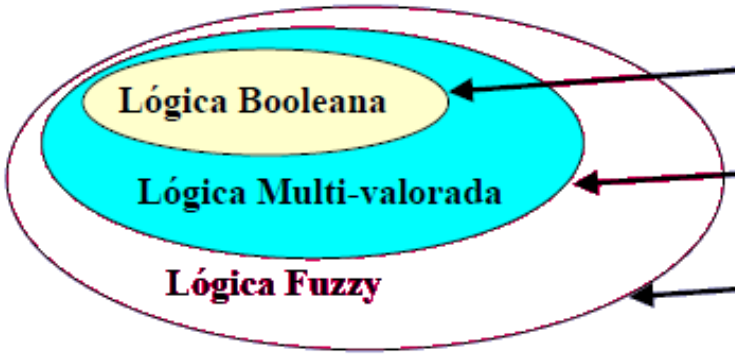
Classificação com regras de decisão crisp:

```
NDVI=imread('ndvi.tif');
MDS=imread('MDSn_metros.tif');
[m,n]=size(MDS)
Classe = zeros(m,n);

% 1 = gramado
% 2 = arvore
% 3 = solo, asfalto
% 4 = telhado

for i = 1: m
for j = 1:n
      if MDS(i,j) <2 & NDVI(i,j) > 90
                  Classe(i,j) = 1;
      elseif MDS(i,j) >2 & NDVI(i,j) > 90
                  Classe(i,j) = 2;
      elseif MDS(i,j) <2 & NDVI(i,j) < 90
                  Classe(i,j) = 3;
      elseif MDS(i,j) >2 & NDVI(i,j) < 90
                  Classe(i,j) = 4;
            end
end

result = uint8(Classe);
imwrite(Classe,'classes.tif')
```



- A Lógica Nebulosa consiste de um conjunto de princípios matemáticos para a representação do conhecimento com base em ***graus de pertinência*** em vez da pertinência da *lógica clássica* binária.

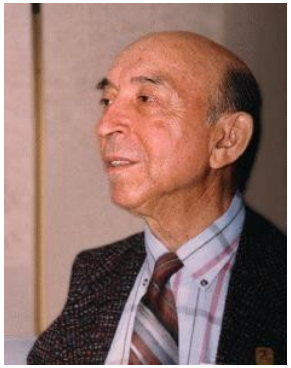
- A lógica nebulosa
  - É multi-valorada.
  - Lida com graus de pertinência e **graus de verdade**.
    - Os valores lógicos estão entre 0 (completamente falso) e 1 (completamente verdadeiro).



Boole: verdadeiro/falso

Lukasiewicz: representa o valor indefinido

Zadeh: quantifica o grau de certeza/incerteza para modelar processos de decisão



Lotfi **Zadeh**, o “pai” da lógica fuzzy.

Conjunto fuzzy

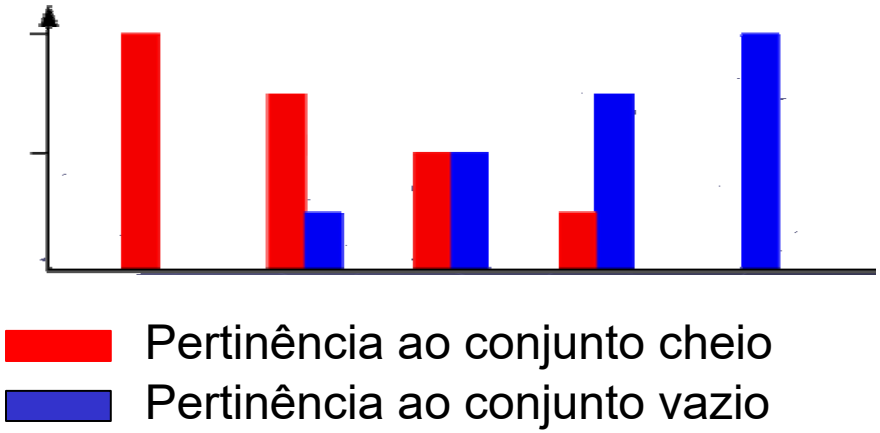
- Um conjunto fuzzy A do universo X é definido pela função $\mu_A(x)$, denominada **função de pertinência** do conjunto A.

$\mu_A(x) : X \longrightarrow [0; 1]$

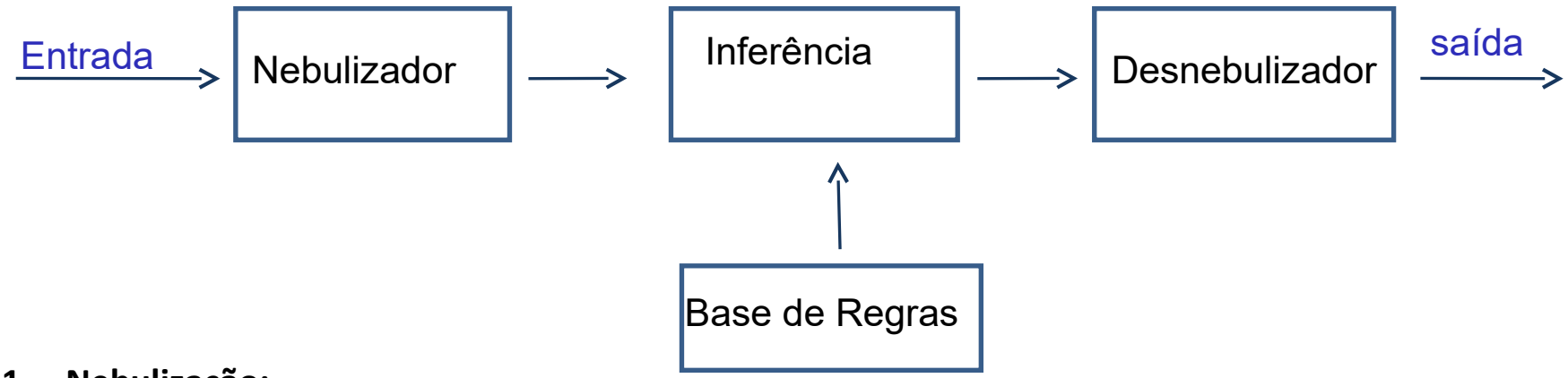
sendo:

- $\mu_A(x) = 1$ se x estiver completamente em A
- $\mu_A(x) = 0$ if x não estiver em A;
- $0 < \mu_A(x) < 1$ se x estiver parcialmente em A

A função de pertinencia μ A(x) indica a certeza que temos em que um dado elemento x pertence ao conjunto A



Sistema de inferência Fuzzy:



1. **Nebulização:**

   Nesta etapa, o problema é analisado e os dados de entrada são transformados em variáveis linguísticas.

-> Como geralmente os dados de entrada são valores *crisp, resultados de medições ou observações, é necessário efetuar-se o* mapeamento destes dados para os conjuntos nebulosos de entrada relevantes.

2. **Inferência:**

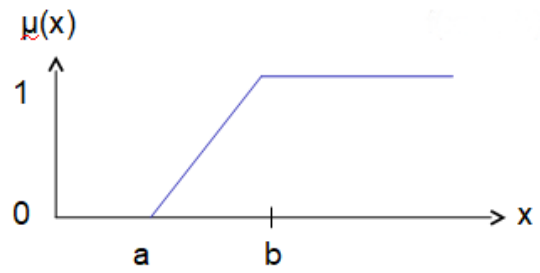
   Nesta etapa serão :(a) avaliadas as regras ou proposições através da associação das variáveis, e (b) serão agregadas as consequentes das regras.

-> Ocorrem as operações com conjuntos nebulosos propriamente ditas.

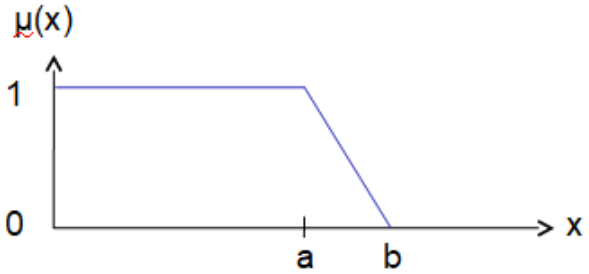
3. **Desnebulização:**

   É a etapa em que os valores *fuzzy são* convertidos em números reais tendo assim valores de saída *crisp.*

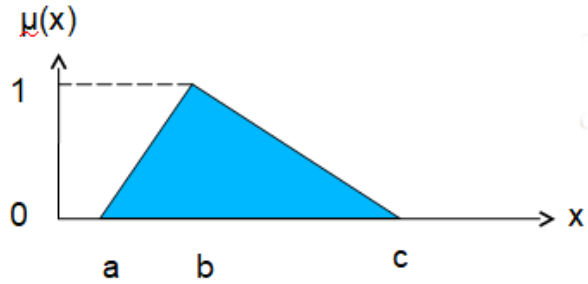
Funções de pertinência



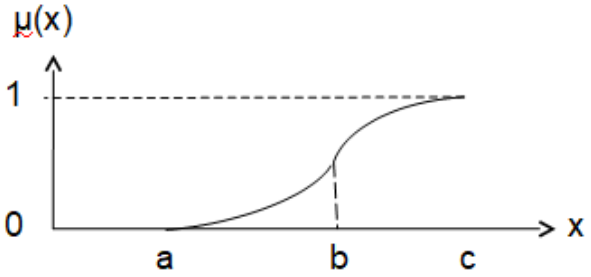
$$f(x; a, b) = \begin{cases} 0 & \text{se } x < a \\ (x-a) / (b-a) & \text{se } a \le x < b \\ 1 & \text{se } b \le x \end{cases}$$



$$f(x; a, b) = \begin{cases} 1 & \text{se } x \le a \\ (b-x) / (b-a) & \text{se } a < x < b \\ 0 & \text{se } b \le x \end{cases}$$



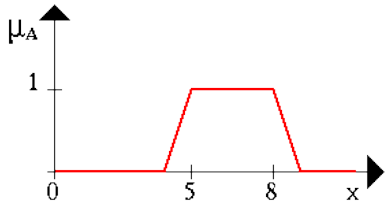
$$f(x; a, b, c) = \begin{cases} 0 & \text{se } x < a \\ (x-a) / (b-a) & \text{se } a \le x < b \\ (c-x) / (c-b) & \text{se } b \le x < c \\ 0 & \text{se } c \le x \end{cases}$$

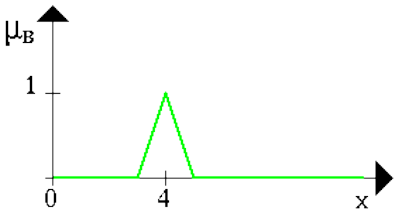


$$f(x; a,b,c) = \begin{cases} 0 & \text{se } x < a \\ 2\,((x-a) / (c-a))^2 & \text{se } a \le x < b \\ 1 - 2((x-c) / (c-a))^2 & \text{se } b < x \le c \end{cases}$$

Operadores com Conjuntos *Fuzzy*

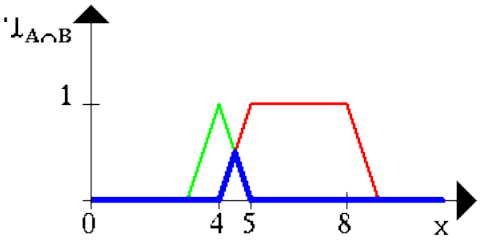
Dados os conjuntos fuzzy A e B:





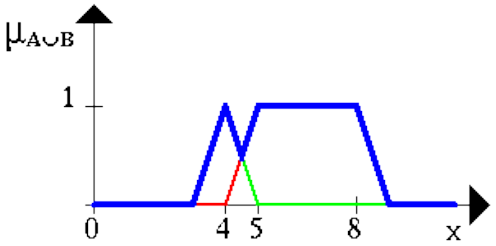
- Intersecção (AND):

$$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \ \forall x \in X$$



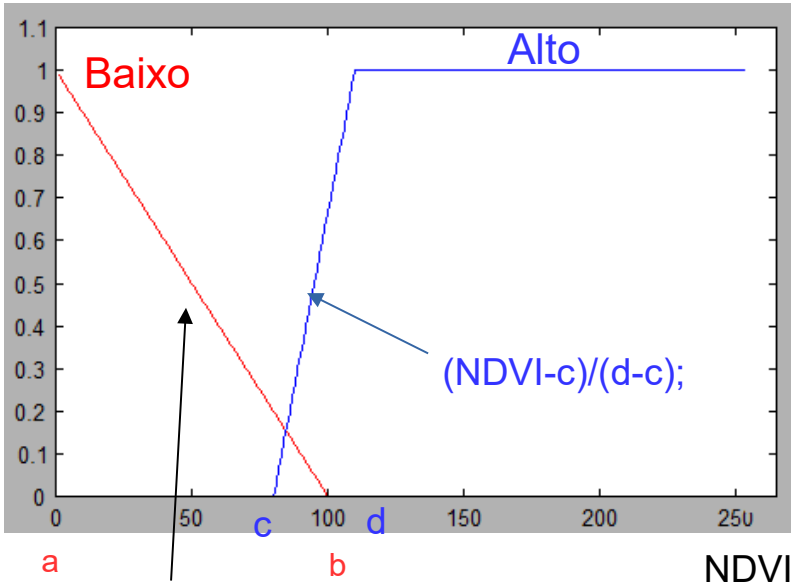
- União (OR):

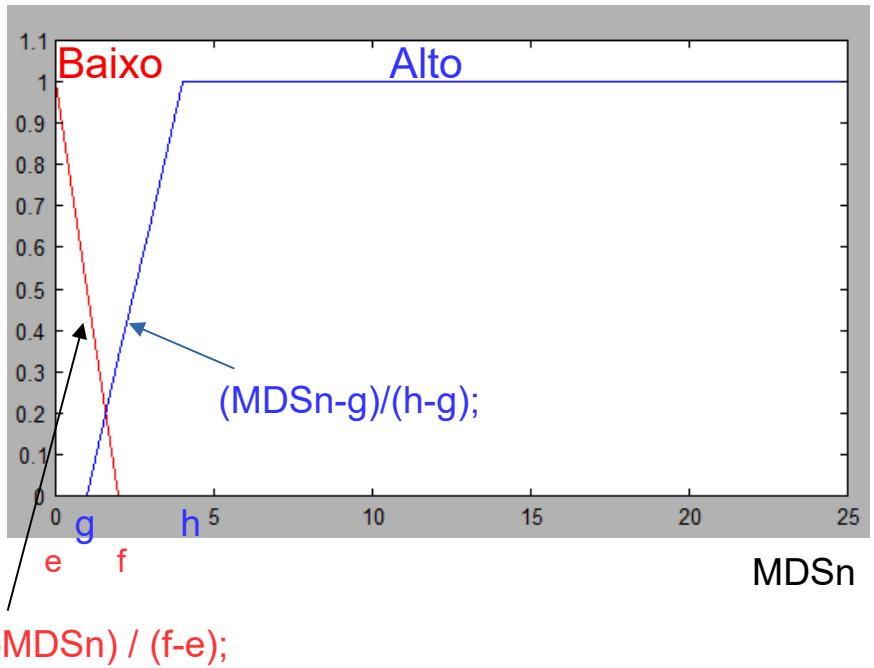
$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \ \forall x \in X$$



Inferência fuzzy simplificada para o caso de saídas discretas (classes c_1 , c_2, … c_n)

Funções de pertinência e conjuntos:





Aplicando as regras:

$$m_1 = \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0))$$
$$m_2 = \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0))$$
$$\vdots$$
$$m_n = \min(\mu_{A_n}(x_0), \mu_{B_n}(y_0))$$

Onde x_0 = NDVI (i,j);
y_0 = MDSn(i,j) ;

Calculando a conclusão para cada regra

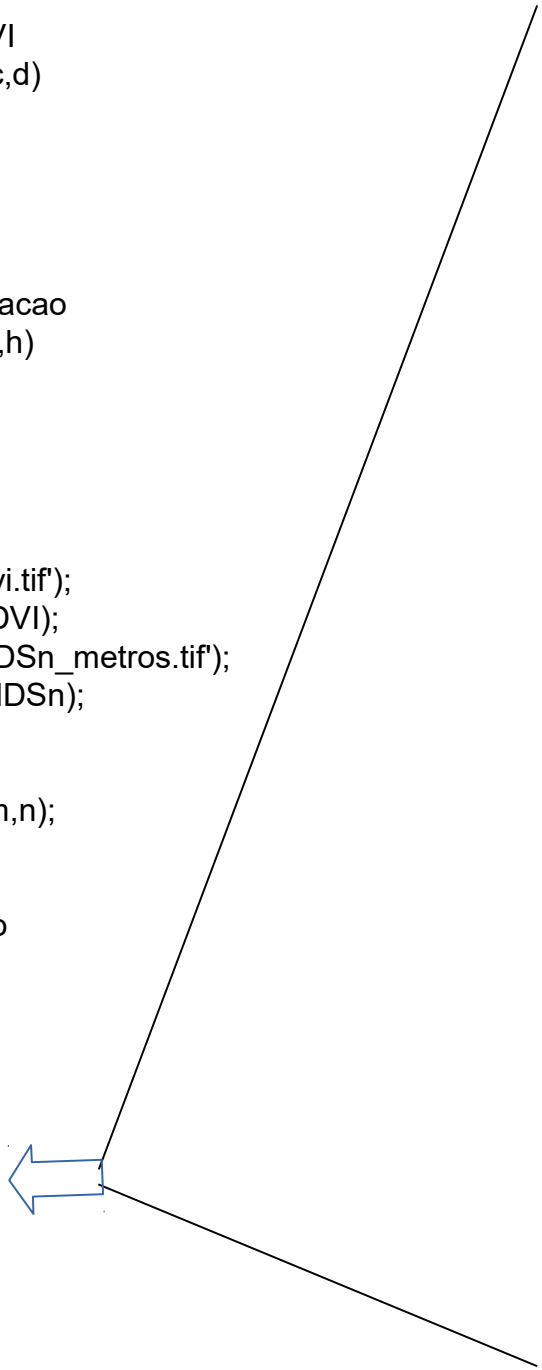
$$c_1' = m_1 \cdot c_1$$
$$c_2' = m_2 \cdot c_2$$
$$\vdots$$
$$c_n' = m_n \cdot c_n$$

Calculando a conclusão final:

$$c' = \frac{\sum_{i=1}^{n} c_i'}{\sum_{i=1}^{n} m_i}$$

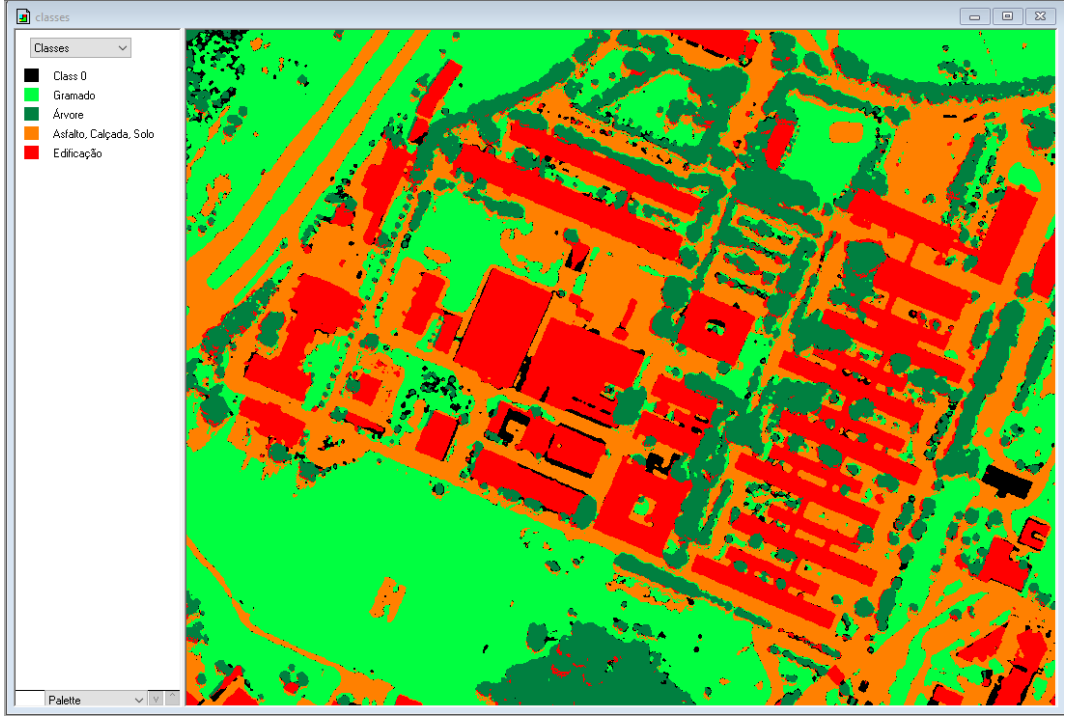
```
% Variável: NDVI
% definir (a,b),(c,d)
a = 0
b = 100
c = 80
d = 110

% Variável: elevacao
% definir (e,f),(g,h)
e = 0
f = 2
g = 1
h = 4

NDVI=imread('ndvi.tif');
NDVI = double(NDVI);
MDSn=imread('MDSn_metros.tif');
MDSn = double(MDSn);
[m,n]=size(MDSn)

Classes = zeros(m,n);
% 1 = gramado
% 2 = arvore
% 3 = solo, asfalto
% 4 = telhado

    for i = 1:m
    for j = 1:n

    end
    end
```

```
% Variável: NDVI
% definir (a,b),(c,d)
a = 0
b = 100
c = 80
d = 110

% Variável: elevacao
% definir (e,f),(g,h)
e = 0
f = 2
g = 1
h = 4

NDVI=imread('ndvi.tif');
NDVI = double(NDVI);
MDSn=imread('MDSn_metros.tif');
MDSn = double(MDSn);
[m,n]=size(MDSn)

Classes = zeros(m,n);
% 1 = gramado
% 2 = arvore
% 3 = solo, asfalto
% 4 = telhado

    for i = 1:m
    for j = 1:n

    end
    end
```



```
ndvi = NDVI(i,j);
if ndvi < b
        ndvi_baixo = (b-ndvi) / (b-a);
end
if ndvi >b
        ndvi_baixo = 0;
end
if ndvi <= c
        ndvi_alto = 0;
end
if ndvi > c & ndvi <= d
        ndvi_alto = (ndvi-c)/(d-c);
end
if ndvi  > d
        ndvi_alto = 1;
end
altura = MDSn(i,j);
if altura <= f
        altura_baixo = (f-altura) / (f-e);
end
if altura > f
        altura_baixo = 0;
end
if altura <= g
        altura_alto = 0;
end
if altura > g & altura <= h
        altura_alto = (altura-g)/(h-g);
end
if altura >= h
        altura_alto = 1;
end
% Cálculo da conclusão de cada regra
r1 = min(altura_baixo, ndvi_alto );
r2 = min(altura_alto, ndvi_alto);
r3 = min( altura_baixo, ndvi_baixo );
r4 = min( altura_alto, ndvi_baixo);
% Conclusão final para o pixel (i,j)
Classes(i,j) = round( (r1*1 + r2*2 + r3*3 + r4*4) / (r1+r2+r3+r4) );
```

Classificação com base em regras *crisp*



Classificação com base em regras *fuzzy*